

What is claimed is:

1. A process for supplementing pre-prepared media digital file content to be performed by a digital playback apparatus with supplemental digital program content, that comprises, preparing such supplemental digital program content in the form of executable code; and embedding the executable code into the pre-prepared media file for execution by the playback apparatus supplementary to the playback of the pre-prepared media-file content.
2. The process of claim 1 wherein the media file has not been pre-prepared to contain such executable code, and the code is seamlessly embedded in the media file as supplemental digital program sequences of executable code.
3. The process of claim 2 wherein the media digital file program content has been pre-prepared from the group consisting of audio, video, image, 3-D, database information and combinations thereof.
4. The process of claim 3 wherein the sequences of executable code are prepared in any computer program format.
5. The process of claim 4 wherein the sequences of executable code are prepared from the group consisting of computer-programmed Java class files, Macromedia Shockwave, Flash, binary executables, byte codes, Visual Basic and Java Script.

6. The process of claim 4 wherein the program content of said sequences of executable code are selected as one or more of graphic, interactive, and e-commerce content.
7. The process of claim 6 wherein said program content includes one or more of advertising, transactional advertising, interactive music videos, and e-commerce.
8. The process of claim 3 wherein the media file has been pre--prepared to comprise audio program format and the embedded executable code is prepared to provide image supplementation.
9. The process of claim 1 wherein the embedding in the media file is effected at predetermined time intervals.
10. The process of claim 1 wherein the execution of the code is synchronized with the playback of the media files.
11. The process of claim 1 wherein said embedding of the executable code into the media file is effected seamlessly with backward compatibility with the media file format, and is unobtrusive to a user of the playback.

12. The process of claim 11 wherein a checksum is used during the execution of the code at the playback, to verify that the embedded executable code has been extracted correctly.

13. The process of claim 1 wherein the embedding is effected by steganographic techniques.

14. The process of claim 13 wherein the encoding of the executable code in the media file is effected by one of low-bit encoding and frequency-domain low-bit encoding.

15. The process of claim 13 wherein the embedding is effected at substantially greater than 300 bits/second of executable code in the media file.

16. The process of claim 1 wherein the media file is an MPEG audio file containing an MP3 audio stream.

17. The process of claim 16 wherein, in the encoding process, the executable code is unsynchronized from synchronization bytes of the audio stream and encapsulated in ID3v2 format before insertion at the beginning of the MP3 audio stream from the audio file.

18. The process of claim 16 wherein, in the encoding process, executable code bits are inserted at the end of the audio file data in each audio frame to encode ancillary data.

66000" 216350

19. The process of claim 18 wherein the MPEG audio file is recompressed to ensure room for executable code at the end of the frames.

20. The process of claim 16 wherein the executable code is encoded in private data packets for the insertion and embedding of new private data packets into the existing MPEG file.

21. The process of claim 13 wherein the executable code is transformed into a bit stream and inserted and embedded at selected locations in the media file identified as locations where minor changes to the media file content produce minimal effects during playback.

22. The process of claim 1 wherein, prior to the encoding embedding of the executable code into the media file, the media file is subjected to a digital watermarking process.

23. The process of claim 1 wherein the pre-prepared media file is an MP3 file and the executable code is seamlessly embedded therein and prepared to provide the supplemental program content selected from the group consisting of transactional advertising, games, polls, contests, interactive music videos and e-commerce.

24. A system for flexibly adding supplemental digital program content to the playback of a pre-prepared media digital file by digital playback apparatus, comprising, means for modifying the pre-prepared media file to embed sequences of executable code therein

representing such supplemental program content; means provided in the digital playback apparatus for decoding the embedded code during playback of the modified media file at the digital playback apparatus; and, in addition to means for playing back the pre-prepared content of the media file, means provided at the digital playback apparatus responsive to the decoding for also presenting thereat the supplemental program content.

25. The system of claim 24 wherein the executable code sequences are selected to contain one or more of graphic, interactive and e-commerce program content.

26. The system of claim 25 wherein said program content includes one or more of advertising, transactional advertising, games, polls, contests, interactive music videos and e-commerce.

27. The system of claim 24 wherein the media file is an MPEG audio file containing an MP3 audio stream.

28. The system of claim 27 wherein the modifying means comprises means for unsynchronizing the executable code from synchronization bytes of said audio stream and encapsulating the code in ID3v2 format before insertion at the beginning of the MP3 audio stream from the audio file.

29. The system of claim 27 wherein, in the encoding process, means is provided for inserting executable code bits at the end of the audio file data in each audio frame to encode ancillary data.

30. The system of claim 29 wherein the MPEG audio file is recompressed to ensure room for executable code at the end of the frames.

31. The system of claim 27 wherein means is provided for encoding the executable code in private data packets for the insertion and embedding of new private data packets into the existing MPEG file.

32. The system of claim 24 wherein the modifying means comprises steganographic coding means and wherein means is provided for transforming the executable code into a bit stream and for inserting and embedding bits at selected locations in the media file, identified as locations where minor changes to the media file content produce minimal effects during playback.

33. The system of claim 24 wherein means is provided, operable prior to the encoding-embedding of the executable code into the media file, for subjecting the media file to a digital watermarking process.

34. The system of claim 24 wherein the pre--prepared media file is an MP3 file and the executable code is seamlessly embedded therein and prepared to provide supplemental

program content selected from the group consisting of transactional advertising, games, polls, contests, interactive music videos and e-commerce.

35. A method of conducting advertising and e-commerce business through an expanded use of digital media playing apparatus, that comprises, seamlessly embedding in digital entertainment media files pre-prepared for entertainment playback by said apparatus, executable code representing supplementary digital advertising and e-commerce business solicitation program content; and modifying said apparatus to enable also decoding of said code by the apparatus so as to enable playback at said apparatus of said business solicitation program content as a supplement to the playback of the entertainment file, and without affecting the backwards compatibility of the file format, thereby to provide business solicitation opportunities not previously provided at such apparatus.

36. The method of claim 35 wherein the media file contains audio MPEG formats and said playback apparatus is adapted for playing MP3 data streams of such formats.

add a' >